




LARANA, INC.

SEMESTA HACKATHON

OLEH SILVI KUSUMA WARDHANI GUNAWAN



CONTENT

- 
- | | |
|-----------|----------------------------|
| 01 | LOGIN, NAVIGASI, DAN INDEX |
| 02 | HOME |
| 03 | JADWAL |
| 04 | TUGAS |
| 05 | CATATAN |
| 06 | ASISTEN AI |
| 07 | CONTACT |

LOGIN

1. Ketika pengguna mengisi formulir dengan NIK dan nama lengkap, kemudian menekan tombol "Murid baru" atau "Masuk", data dikirimkan melalui metode POST ke skrip PHP.
2. Jika pengguna memilih tombol "Murid baru" (pengguna baru), skrip PHP akan memeriksa apakah file dengan nama yang sesuai (berdasarkan NIK dan nama) sudah ada. Jika belum ada, skrip akan membuat file baru dengan nama tersebut di direktori "file/" dan menyimpannya kosong. Informasi pengguna (NIK dan nama) juga disimpan dalam variabel session. Pengguna kemudian akan diarahkan kembali ke halaman utama setelah 2 detik.
3. Jika pengguna memilih tombol "Masuk", skrip PHP akan memeriksa apakah file dengan nama yang sesuai (berdasarkan NIK dan nama) ada. Jika ada, informasi pengguna (NIK dan nama) disimpan dalam variabel session. Pengguna kemudian akan diarahkan kembali ke halaman utama setelah 2 detik.

LOGIN

4. Jika file dengan nama yang sesuai tidak ditemukan dalam kedua kasus di atas, skrip akan menampilkan pesan kesalahan yang sesuai.
5. Halaman HTML menggunakan Bootstrap untuk tampilan yang lebih menarik. Terdapat formulir dengan input untuk NIK dan nama, serta dua tombol "Murid baru" dan "Masuk".
6. File CSS dan JavaScript digunakan untuk mengatur tampilan dan menghubungkan dengan library Bootstrap.

PINTARKU

Asisten murid terbaik



Murid baru

Masuk

INDEX

1. Skrip PHP pada awal file digunakan untuk memulai sesi dan mengambil nilai NIK dan nama dari sesi yang sedang berjalan. Jika sesi pengguna belum ada atau tidak valid, pengguna akan diarahkan kembali ke halaman login.
2. Terdapat layout yang terdiri dari dua bagian, yaitu navigasi di kolom kiri (**col-lg-3**) dan konten utama di kolom kanan (**col-lg-9**).
3. Di dalam kolom kiri, terdapat file **navigasi.php** yang akan di-include. File ini berisi navigasi atau menu untuk mengakses berbagai halaman atau fitur dalam aplikasi.
4. Di dalam kolom kanan, terdapat logika yang memeriksa nilai dari variabel **\$page** yang diterima melalui parameter GET. Berdasarkan nilai **\$page**, akan di-include file yang sesuai untuk menampilkan halaman atau konten yang relevan. Pilihan halaman termasuk "asistenAI", "jadwal", "home" (default jika **\$page** kosong), "catatan", "contact", "tugas", dan "logout".
5. Jika pengguna memilih halaman "logout", sesi pengguna akan dihapus dan pengguna akan diarahkan kembali ke halaman login.

NAVIGASI

1. Struktur HTML menggunakan Bootstrap untuk membuat tampilan kartu (`<div class="card">`) yang berisi logo aplikasi.
2. Di dalam kartu, terdapat elemen gambar (``) yang menampilkan logo aplikasi. Logo tersebut memiliki atribut `src` yang mengarahkan ke file `logo.png`, dan atribut `width` dan `height` untuk mengatur ukuran logo.
3. Setelah kartu logo, terdapat daftar navigasi yang ditampilkan dalam format daftar grup (`<div class="list-group">`). Setiap item navigasi adalah tautan (`<a>`) yang mengarahkan pengguna ke halaman atau fitur yang sesuai.
4. Setiap tautan navigasi memiliki atribut `href` yang mengandung URL dengan parameter GET `page` yang sesuai dengan halaman atau fitur yang ingin diakses. Misalnya, `?page=home` mengarahkan pengguna ke halaman "home".
5. Untuk menentukan tampilan aktif (diberi penanda sebagai aktif), terdapat penggunaan logika PHP. Dalam contoh ini, jika nilai variabel `$page` sama dengan halaman yang sedang aktif atau jika variabel `$page` kosong, maka tautan tersebut diberikan kelas CSS "active" untuk menandakan status aktif.
6. Terdapat juga tautan "Logout" yang tidak membutuhkan penanda aktif. Ketika tautan ini diklik, pengguna akan diarahkan ke halaman "logout" yang akan menghapus sesi pengguna dan mengarahkan pengguna kembali ke halaman login.

HOME

1. Terdapat tiga elemen `<div>` dengan kelas "alert alert-primary" yang digunakan untuk menampilkan pesan selamat datang dan semangat kepada pengguna.
 2. Pada elemen pertama, terdapat teks "Selamat Datang" yang diikuti oleh variabel `$nama` yang menampilkan nama pengguna yang disimpan dalam sesi sebelumnya. Pesan ini memberikan sambutan kepada pengguna dan menyebutkan nama mereka.
 3. Pada elemen kedua, terdapat variabel `$nama` yang ditampilkan kembali, diikuti dengan pesan semangat untuk meraih mimpi dan semangat belajar. Pesan ini memberikan dorongan motivasi kepada pengguna.
 4. Pada elemen ketiga, terdapat pesan yang menyatakan bahwa "Website Pintarku" adalah asisten belajar pengguna. Pesan ini memberikan penjelasan singkat tentang peran dan tujuan dari website tersebut.
- File "home.php" bertanggung jawab untuk menampilkan pesan selamat datang, semangat belajar, dan penjelasan singkat tentang asisten belajar dalam halaman beranda aplikasi. Pesan-pesan tersebut memberikan penggunaan pengalaman yang positif dan memotivasi untuk menggunakan fitur-fitur aplikasi yang tersedia.

PINTARKU
Asisten murid terbaik



Home

Asisten

Jadwal

Tugas

Catatan

Contact

Logout

Selamat Datang silvi Di Website Pintarku

silvi Semangat Meraih Mimpi Kalian! Semangat Belajar!

Bersama Website Pintarku Asisten Belajar Anda

JADWAL

1. Terdapat elemen `<table>` yang digunakan untuk menampilkan jadwal sekolah dalam bentuk tabel.
 2. Pada elemen `<tr>` pertama, terdapat `<th>` yang menampilkan judul kolom tabel, yaitu "Hari", "Mata Pelajaran", dan "Jam".
 3. Terdapat variabel `$jadwal` yang berisi array multidimensi yang menyimpan jadwal sekolah. Setiap hari memiliki array mata pelajaran yang terdiri dari nama mata pelajaran dan jam pelajaran.
 4. Terdapat pengulangan `foreach` yang digunakan untuk mengakses setiap hari dan mata pelajaran pada variabel `$jadwal`.
 5. Pada setiap iterasi pengulangan, dilakukan penghitungan jumlah pelajaran pada hari tersebut dengan menggunakan fungsi `count()`. Nilai tersebut digunakan untuk menentukan atribut `rowspan` pada elemen `<td>` yang menampilkan nama hari.
 6. Terdapat pengulangan `foreach` kedua yang digunakan untuk mengakses setiap mata pelajaran pada hari tersebut. Mata pelajaran dan jam pelajaran ditampilkan dalam elemen `<td>`.
 7. Setelah itu, ditutup elemen `<tr>` untuk setiap mata pelajaran pada hari tersebut.
 8. Setelah pengulangan selesai, tabel jadwal sekolah telah selesai ditampilkan.
- File "jadwal.php" bertanggung jawab untuk menampilkan jadwal sekolah dalam bentuk tabel dengan menggunakan data yang disimpan dalam variabel `$jadwal`. Jadwal sekolah tersebut terdiri dari nama hari, mata pelajaran, dan jam pelajaran. Tabel jadwal sekolah ini memudahkan pengguna untuk melihat jadwal pelajaran mereka secara terstruktur.

[Home](#)[Asisten](#)[Jadwal](#)[Tugas](#)[Catatan](#)[Contact](#)[Logout](#)

Jadwal Sekolah

| Hari | Mata Pelajaran | Jam |
|-------|----------------|---------------|
| Senin | Matematika | 08:00 - 09:00 |
| | Bahasa Inggris | 09:00 - 10:00 |
| | Fisika | 10:00 - 11:00 |
| | | |

TUGAS

1. Terdapat logika PHP yang dijalankan ketika tombol "Simpan" pada form ditekan. Data yang diisi pada form, seperti tanggal, jam, mata pelajaran, jenis tugas, dan catatan, akan disimpan.
 2. Data tugas siswa disimpan dalam file teks dengan nama file yang didasarkan pada variabel **\$nik** dan **\$nama**. File akan disimpan dalam folder "file/". Jika file sudah ada (baris pertama sudah terisi), data tugas akan ditambahkan pada baris baru. Jika file masih kosong, data tugas akan ditulis pada baris pertama.
 3. Pada bagian HTML, terdapat sebuah form yang menggunakan metode POST untuk mengirim data ke halaman ini sendiri.
 4. Terdapat elemen-elemen input seperti **<input type="date">**, **<input type="time">**, **<input type="text">**, dan elemen radio button untuk memasukkan data tugas siswa.
 5. Tombol "Simpan" digunakan untuk mengirimkan data tugas ke server untuk disimpan.
 6. Tombol "Reset" digunakan untuk menghapus data yang telah dimasukkan pada form.
- File "tugas.php" bertanggung jawab untuk menampilkan form agenda siswa yang digunakan untuk memasukkan data tugas siswa. Data tugas yang dimasukkan akan disimpan dalam file teks lalu dimasukkan ke "catatan.php" untuk tampilkan dalam bentuk tabel



Home

Asisten

Jadwal

Tugas

Catatan

Contact

Logout

FORM AGENDA SISWA

Tanggal

hh/bb/tttt



Jam

--:--



Mata Pelajaran

Tugas Jenis Tugas

- ☐ PR
- ☐ Tugas Kelompok
- ☐ Latihan Soal

Catatan Agenda Hari ini

Simpan

Reset

CATATAN

1. File ini menggunakan class "card" dari Bootstrap untuk mengatur tampilan kartu.
 2. Terdapat judul "AGENDA HARI INI" yang ditampilkan pada header kartu.
 3. Tabel dengan ID "data-table" dibuat untuk menampilkan data agenda.
 4. Pada bagian "thead" (header tabel), terdapat kolom-kolom yang akan ditampilkan, yaitu Tanggal, Waktu, Mata Pelajaran, Tugas, dan Catatan.
 5. Pada bagian "tbody" (isi tabel), terdapat logika PHP yang membaca isi file teks berdasarkan variabel **\$nik** dan **\$nama**. Setiap baris dalam file teks dipecah menjadi array menggunakan delimiter "|".
 6. Setiap baris data agenda dalam file teks akan ditampilkan dalam baris tabel. Data seperti tanggal, waktu, mata pelajaran, tugas, dan catatan akan ditampilkan dalam kolom-kolom yang sesuai.
 7. Pengulangan **while(!feof(\$fh))** digunakan untuk membaca baris demi baris dalam file teks sampai mencapai akhir file.
 8. Tabel ini bersifat responsif dan dapat menggunakan fitur pengurutan dan pencarian jika menggunakan plugin DataTables.
- File "catatan.php" bertanggung jawab untuk menampilkan tabel yang berisi agenda hari ini berdasarkan data yang tersimpan dalam file teks. Tabel ini memberikan gambaran cepat tentang tugas dan catatan yang perlu dikerjakan atau diperhatikan pada hari tersebut.

[Home](#)[Asisten](#)[Jadwal](#)[Tugas](#)[Catatan](#)[Contact](#)[Logout](#)

AGENDA HARI INI

Show 10 entries

Search:

| Tanggal | Waktu | Mata Pelajaran | Tugas | Catatan |
|------------|-------|----------------|--------------|-------------|
| 2023-06-25 | 07:53 | matematika | Latihan_soal | latihan uas |

Showing 1 to 1 of 1 entries

Previous 1 Next

ASISTEN AI

1. Fungsi **asistenBelajarAI(\$pertanyaan)**: Ini adalah fungsi yang menghubungkan ke API GPT-3.5 Turbo dari OpenAI untuk meminta jawaban dari pertanyaan yang diberikan. Fungsi ini mengirim permintaan ke endpoint API dengan menggunakan cURL dan mengembalikan jawaban yang diberikan oleh model AI.
2. Variabel **\$openai_key**: Ini adalah kunci API OpenAI yang digunakan untuk mengotentikasi permintaan ke API GPT-3.5 Turbo. Pastikan untuk mengganti nilai ini dengan kunci API yang valid.
3. Blok kode **if (\$_SERVER['REQUEST_METHOD'] === 'POST')**: Ini adalah blok kode yang dipicu ketika formulir dikirim dengan metode POST. Pertanyaan dari formulir akan diambil dan fungsi **asistenBelajarAI()** akan dipanggil untuk mendapatkan jawaban dari pertanyaan tersebut.
4. Bagian HTML: File ini juga berisi tampilan HTML sederhana yang mencakup sebuah formulir untuk mengirim pertanyaan ke asisten AI dan menampilkan log percakapan antara pengguna dan asisten AI.
5. Di dalam tag **<style>**: Terdapat beberapa aturan CSS sederhana untuk mengatur tampilan elemen-elemen dalam halaman.
6. Di dalam tag **<form>**: Terdapat formulir yang mengandung input teks dengan nama "pertanyaan" di mana pengguna dapat memasukkan pertanyaan mereka. Ketika formulir dikirim, pertanyaan akan dikirim ke server untuk diolah.

ASISTEN AI

7. Setelah formulir dikirim, jika variabel **\$jawaban** diatur (artinya telah menerima respons dari API AI), maka pertanyaan dan jawaban akan ditampilkan dalam elemen **<div>** dengan kelas "user-message" di bawah formulir.

8. File ini mengandalkan API GPT-3.5 Turbo dari OpenAI untuk memberikan jawaban atas pertanyaan yang diberikan. Pastikan Anda memiliki koneksi internet yang aktif dan menggunakan kunci API yang valid agar fungsi ini dapat berfungsi dengan benar.

File "asisten_belajar_ai.php" memungkinkan pengguna untuk berinteraksi dengan asisten AI sederhana dengan mengajukan pertanyaan dan menerima jawaban dari model AI melalui API GPT-3.5 Turbo khususnya para user yang merupakan murid jika adanya open ai di web ini akan mempermudah mencari informasi untuk belajar ataupun untuk tugas siswa

Home

Asisten

Jadwal

Tugas

Catatan

Contact

Logout

Asisten Belajar AI

Ketik pertanyaan di sini

Tanyakan

CONTACT

1. Bagian HTML: File ini mengandung tampilan HTML sederhana yang mencakup elemen-elemen berikut:

- Judul halaman: Menggunakan tag **<title>**.
- Gaya CSS: Menggunakan tag **<style>** untuk mendefinisikan gaya tampilan elemen-elemen dalam halaman.
- Kontainer utama: Menggunakan elemen **<div>** dengan kelas "container" untuk mengelompokkan semua elemen dalam halaman.
- Tombol "Chat dengan Guru" dan "Chat dengan Layanan Pelanggan": Menggunakan elemen **<button>** dengan atribut **onclick** yang memanggil fungsi **openChat()** dengan argumen yang sesuai (guru atau customer_service).
- Kotak chat: Menggunakan elemen **<div>** dengan kelas "chatbox" dan ID "chatbox". Ini adalah area tempat pesan ditampilkan.
- Header chat: Menggunakan elemen **<div>** dengan kelas "chat-header" dan atribut **onclick** yang memanggil fungsi **toggleChatbox()**. Ketika diklik, ini akan menyembunyikan atau menampilkan kotak chat.
- Pesan-pesan chat: Menggunakan elemen **<div>** dengan kelas "chat-messages" dan ID "chat-messages". Ini adalah area di mana pesan-pesan ditampilkan.
- Input pesan: Menggunakan elemen **<input>** dengan tipe "text" dan ID "message-input" untuk memasukkan pesan yang akan dikirim.
- Tombol "Kirim": Menggunakan elemen **<input>** dengan tipe "submit" dan atribut **onclick** yang memanggil fungsi **sendMessage()**. Ketika tombol ini diklik, pesan akan dikirim.
- Kotak dukungan: Menggunakan elemen **<div>** dengan kelas "support-box" untuk menampilkan area bantuan dan dukungan.
- Textarea pertanyaan/ masalah: Menggunakan elemen **<textarea>** untuk memasukkan pertanyaan atau masalah pengguna.
- Tombol "Kirim" (dukungan): Menggunakan elemen **<input>** dengan tipe "submit" untuk mengirim pertanyaan atau masalah pengguna.

2. Fungsi **openChat(role)**: Fungsi ini dipanggil ketika tombol "Chat dengan Guru" atau "Chat dengan Layanan Pelanggan" diklik. Fungsi ini akan menampilkan kotak chat, membersihkan pesan-pesan sebelumnya, dan menambahkan pesan sambutan sesuai dengan peran yang diberikan (guru atau customer_service).

CONTACT

1. Fungsi **toggleChatbox()**: Fungsi ini dipanggil ketika header chat diklik. Fungsi ini akan menyembunyikan atau menampilkan kotak chat dengan mengubah properti **display** CSS.
2. Fungsi **sendMessage()**: Fungsi ini dipanggil ketika tombol "Kirim" diklik. Fungsi ini akan mengambil teks dari input pesan, membersihkan input pesan, dan memanggil fungsi **appendMessage()** untuk menambahkan pesan pengguna ke daftar pesan chat.
3. Fungsi **appendMessage(sender, message)**: Fungsi ini digunakan untuk menambahkan pesan ke daftar pesan chat. Fungsi ini membuat elemen-elemen HTML yang sesuai untuk menampilkan pengirim (guru atau pengguna) dan pesan yang dikirim. Elemen-elemen ini kemudian ditambahkan ke daftar pesan chat, dan kotak chat akan otomatis menggulir ke bagian paling bawah untuk menampilkan pesan terbaru.

File "contact.php" digunakan pengguna atau siswa untuk berkomunikasi dengan guru untuk konsultasi tugas dan berkomunikasi dengan layanan pelanggan melalui kotak chat yang disediakan. Pengguna dapat mengirim pesan dan melihat respons dari guru atau layanan pelanggan dalam kotak chat.

Chat dengan Guru dan Layanan Pelanggan

Chat dengan Guru

Chat dengan Layanan Pelanggan

Bantuan dan Dukungan

Tuliskan pertanyaan atau masalah Anda

Kirim

**SEKIAN TERIMA
KASIH**

